

SISTEM PENDUKUNG KEPUTUSAN UNTUK MENENTUKAN POSISI PEMAIN IDEAL DALAM TIM SEPAK BOLA DENGAN METODE ALGORITMA GENETIKA

Hizkia Alpriansa, Anton Setiawan Honggowibowo, Yuliani Indrianingsih
Teknik Informatika STTA Yogyakarta
informatika@stta.ac.id

ABSTRACT

So far, there are coaches who are less precise in determining the ideal position of the player as it only relies on instinct and the ego of the players so that there is still a coach who has not been able to objectively assess the players.

By utilizing the method of Genetic Algorithm as Decision Support System (DSS) in the process of determining the ideal position of a player who uses several criteria (multicriteria) to choose a proper player. DSS is helping coach in making the right decisions and Genetic Algorithm is used as a model for multicriteria weighting in the selection process. This application was built with tools Borland Delphi (7.0) as the user interface design and media processing PostgreSQL as its database.

Based on these results we can conclude that this application expected to assist the coaches in the decision making process and can change the appraisal of which are subjective to more objective, to determine the ideal position for a player, can determine the best position of each position of a number of players and the expected results of the Genetic Algorithm on the system constructed in accordance with the results of manual calculations.

Keywords: Football, Decision Support Systems, Genetic Algorithms

1. PENDAHULUAN

Kualitas pemain sangat menentukan kekuatan suatu tim. Oleh karena itu, dibutuhkan suatu keahlian untuk memilih pemain yang mempunyai kualitas yang sangat bagus. Akan sangat sulit bagi suatu tim untuk memilih pemain, jika tidak ada patokan yang tepat dalam memilih.

Selain itu, dibutuhkan kejelian dalam menilai kemampuan seorang pemain, sehingga layak untuk bergabung dengan tim dan dapat mengangkat prestasi tim tersebut. Tidak jarang terjadi faktor *like* dan *dislike* menjadi penentu dalam memilih pemain, bukan berdasarkan kemampuan pemain sebenarnya. Hal tersebut terkadang membuat pelatih atau manajer tim salah memilih pemain dan setelah beberapa lama pemain tersebut dicampakkan begitu saja karena tidak menunjukkan permainan yang bagus. Hal tersebut sangat disayangkan, apalagi jika harga pemain tersebut sangat mahal, sehingga untuk mendapatkannya membutuhkan pengeluaran yang sangat besar, tetapi pemain tersebut tidak menunjukkan hasil yang sebanding.

Banyak cara atau algoritma untuk mengimplementasikan permasalahan tersebut menjadi suatu aplikasi yang dapat menilai dan memilih pemain yang tepat, sehingga dapat membantu pelatih atau manajer tim untuk mencari pemain yang terbaik. Algoritma Genetika merupakan

suatu metode penyelesaian masalah yang didasarkan pada metode heuristik. Algoritma Genetika adalah metode adaptif yang dapat digunakan untuk mengatasi masalah pencarian dan optimasi. Algoritma Genetika didasarkan pada proses genetika yang dialami organisme dalam evolusi biologi. Kebanyakan generasi, populasi pada alam berkembang sesuai dengan prinsip-prinsip dari seleksi alam, prinsip ini pertama kali secara jelas dinyatakan oleh Charles Darwin dalam "*The Origin of Species*". Dengan menirukan proses ini, algoritma genetika mampu mengembangkan solusi untuk masalah-masalah dalam dunia nyata jika masalah tersebut dapat disandikan.

2. LANDASAN TEORI

2.1 Algoritma Genetika

Algoritma Genetika adalah algoritma yang memanfaatkan proses seleksi alamiah yang dikenal dengan proses evolusi. Simulasi dari proses evolusi Darwin. Teknik pencarian dilakukan sekaligus atas sejumlah solusi yang mungkin yang dikenal dengan populasi. Individu yang terdapat dalam satu populasi disebut dengan istilah kromosom. Kromosom ini merupakan suatu solusi yang masih berbentuk simbol. Populasi awal dibangun dengan secara acak, sedangkan populasi berikutnya merupakan hasil evolusi kromosom-kromosom melalui iterasi yang disebut dengan istilah generasi. Pada setiap generasi, kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang disebut dengan fungsi *fitness*. Nilai *fitness* dari suatu kromosom akan menunjukkan kualitas kromosom dalam populasi tersebut. Generasi berikut dikenal dengan istilah anak (*offspring*) terbentuk dari gabungan dua kromosom generasi sekarang yang bertindak sebagai induk (*parent*) dengan menggunakan operator penyilangan (*crossover*). Selain operator penyilangan, suatu kromosom juga dapat dimodifikasi dengan menggunakan operator mutasi. Generasi baru dibentuk dengan menyeleksi nilai *fitness* sehingga ukuran populasi konstan. Setelah beberapa generasi, maka algoritma ini akan konvergen ke kromosom terbaik.

2.2 Seleksi Roda Roulette (*Roulette Wheel Selection*)

Seleksi Orang Tua (*parent*) yang digunakan adalah Seleksi Roda Roulette (*Roulette Wheel Selection*) dimana orang tua dipilih berdasarkan *fitness* mereka. Lebih baik satu kromosom, lebih besar kesempatan terpilih. Probabilitas suatu individu terpilih untuk perkawinan silang sebanding dengan *fitness*nya.

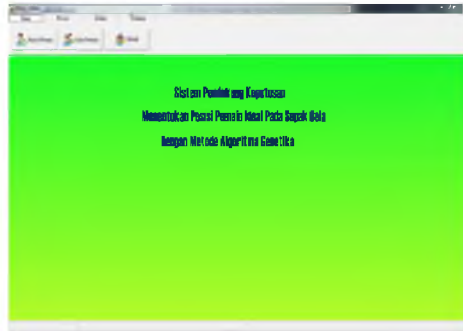
Metode *roulette-wheel* sangat mudah diimplementasikan dalam pemrograman. Pertama, dibuat interval nilai kumulatif (dalam interval $[0,1]$) dari nilai *fitness* masing-masing kromosom dibagi total nilai *fitness* dari semua kromosom. Sebuah kromosom terpilih jika bilangan random yang dibangkitkan berada pada interval nilai akumulatifnya

3. UJI COBA

3.1 Penjelasan Aplikasi

Sesuai rancangan pada gambar 2, didapat hasil penerapan aplikasi sinkronisasi seperti pada gambar 3 dan dapat dijelaskan sebagai berikut :

1. Setelah program ini dijalankan dan berhasil memasukkan *username* dan *password*, maka akan muncul tampilan utama aplikasi ini seperti pada gambar 3.



Gambar 3 Tampilan Menu Utama

2. Langkah selanjutnya adalah dengan memasukkan dan menambah data pemain dengan memilih input pemain.
3. Setelah disimpan akan muncul tampilan posisi pemain yang paling unggul berdasarkan penilaian pelatih sebelumnya dapat dilihat pada gambar 4.

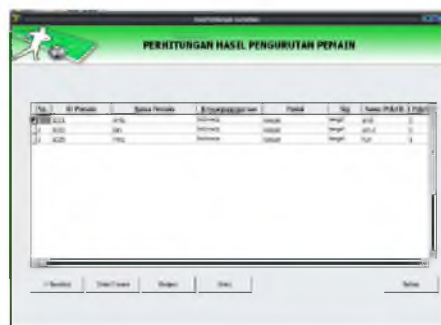


Gambar 4 Tampilan Posisi Pemain

4. Jika ingin melihat data pemain dapat menekan tombol lihat pemain pada menu data.
5. Untuk proses seleksi pemain yang paling unggul berdasarkan posisinya dapat menekan tombol persiapan pada menu proses, pada sub menu persiapan dapat ditentukan posisi dan pemain dari masing-masing pelatih yang akan diseleksi, dapat dilihat pada gambar 5, setelah selesai dipilih, maka tekan tombol lanjutkan untuk memproses posisi yang diinginkan, setelah itu muncul tampilan berdasarkan urutan dari yang paling baik sesuai posisi yang telah ditentukan sebelumnya dapat dilihat pada gambar 6.



Gambar 5, Tampilan Persiapan



No	ID Pemain	Nama Pemain	Bobot	Kecepatan	Keakuratan	Skor	Tingkat
1	101	Andi	85	80	85	150	Andi
2	102	Budi	75	70	75	120	Budi
3	103	Cici	90	90	90	180	Cici

Gambar 6, Tampilan Hasil Pengurutan

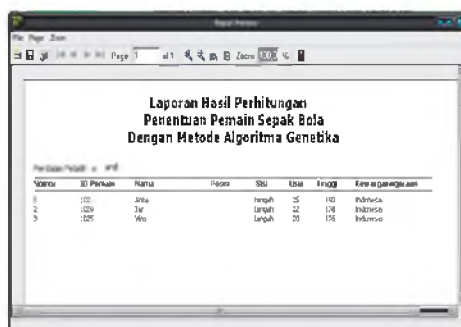
6. Pada tampilan tersebut terdapat tombol detail proses yang berguna untuk melihat proses seleksi yang terjadi ketika menggunakan algoritma genetika, dapat dilihat pada gambar 7.



No	ID Pemain	Nama Pemain	Bobot	Kecepatan	Keakuratan	Skor	Tingkat
1	101	Andi	85	80	85	150	Andi
2	102	Budi	75	70	75	120	Budi
3	103	Cici	90	90	90	180	Cici

Gambar 7, Detail Proses

7. Jika ingin mencetak hasil dari proses seleksi dapat menekan tombol cetak, dapat dilihat pada gambar 8.



No	ID Pemain	Nama	Bobot	Kecepatan	Keakuratan	Skor	Tingkat
1	101	Andi	85	80	85	150	Andi
2	102	Budi	75	70	75	120	Budi
3	103	Cici	90	90	90	180	Cici

Gambar 8, Tampilan Laporan Hasil Seleksi

3.2 Perhitungan Skor

Setiap gen merepresentasikan pemain, dimana setiap pemain mempunyai Skor yang merupakan jumlah total setiap kriteria dikalikan dengan bobot dari setiap kriteria. Jika dituliskan dalam model matematis, yaitu sebagai berikut:

$$SKOR = \sum(w_i * K_i)$$

Dimana:

w_i = bobot kriteria ke-i

K_i = Kriteria ke-i

Setelah ditemukan hasil skor posisi sesuai kriterianya maka dibagi dengan jumlah bobotnya, agar hasil yang didapat objektif. Jika dituliskan dalam model matematis, yaitu sebagai berikut:

$$\text{Skor Posisi} = \frac{\sum \text{SKOR}}{\sum w}$$

Dimana:

SKOR = jumlah hasil dari bobot kriteria ke-i dikali Kriteria ke-i

w = jumlah bobot sesuai kriteria posisi

Fungsi *fitness* yang digunakan untuk mengevaluasi setiap kromosom adalah rata-rata dari jumlah total Skor semua kandidat pemain dikalikan dengan bobotnya. Jika dituliskan dalam model matematis, yaitu sebagai berikut:

$$\text{Fitnees} = \frac{n * SKOR_n + (n - 1) * SKOR_1 + \dots + SKOR_{n-1}}{n}$$

Dimana:

$SKOR_n$ = Skor pemain pada gen ke-0

$SKOR_1$ = Skor pemain pada gen ke-1

$SKOR_{n-1}$ = Skor pemain pada gen terakhir

n = Jumlah Gen

Nilai n, (n-1), sampai dengan 1 berfungsi untuk menentukan urutan dari pemain terbaik, yaitu pada gen pertama sampai pemain terjelek, yaitu pada gen terakhir. Nilai *fitness* akan menunjukkan hasil yang maksimum jika gen-gen dari pertama sampai terakhir memuat pemain dengan Skor terurut dari terbesar sampai yang terkecil.

4. Analisa Hasil Uji Coba

Pada pengujian ini akan dilakukan dua jenis pengujian, yaitu pengujian untuk mengamati hubungan antara masing-masing parameter algoritma genetika terhadap waktu proses dan pengujian untuk mendapatkan *fitness* terbaik.

Pengujian I

Pada pengujian I akan diamati pengaruh parameter ukuran populasi dengan waktu proses. Pada pengujian I ini nilai parameter jumlah maksimum generasi, probabilitas crossover, probabilitas mutasi dan probabilitas elitism akan diset tetap, sedangkan nilai parameter ukuran populasi akan diset bervariasi mulai dari 10 sampai 50.

Tabel 1. Nilai Parameter Tetap pada Pengujian I

Parameter	Nilai Parameter
Jumlah Kandidat	3
Maksimum Generasi	70
Probabilitas Crossover	0,8
Probabilitas Mutasi	0,2

Probabilitas Elistism	0,1
-----------------------	-----

Tabel 4.2. Hasil Pengujian I

Ukuran Populasi	Kromosom Terbaik	<i>Fitness</i>	Waktu (mili detik)
5	2 1 3	3199	64
10	2 3 1	3201	70
15	2 1 3	3199	77
20	2 3 1	3201	90
25	2 3 1	3201	100
30	2 3 1	3201	101
35	2 3 1	3201	104
40	2 3 1	3201	107
45	2 3 1	3201	110
50	2 3 1	3201	124

Dari tabel 2. dapat disimpulkan bahwa perubahan ukuran populasi mempengaruhi waktu proses algoritma genetika yang dilakukan. Pada kolom waktu (mili detik) terlihat bahwa semakin besar nilai ukuran populasi, maka waktu yang dibutuhkan untuk melakukan proses algoritma genetika juga semakin besar.

Pengujian II

Pada pengujian II akan diamati pengaruh parameter jumlah maksimum generasi dengan waktu proses. Pada pengujian II ini nilai parameter ukuran populasi, probabilitas crossover, probabilitas mutasi dan probabilitas elitism akan diset tetap, sedangkan nilai parameter maksimum generasi akan diset bervariasi mulai dari 10 sampai 100. Berikut ini disajikan tabel nilai parameter yang diset tetap yang digunakan pada pengujian II.

Tabel 3 Nilai Parameter Tetap pada Pengujian II

Parameter	Nilai Parameter
Jumlah Kandidat	3
Ukuran Populasi	10
Probabilitas Crossover	0,8
Probabilitas Mutasi	0,2
Probabilitas Elistism	0,1

Tabel 4. Hasil Pengujian II

Maksimum generasi	Kromosom Terbaik	<i>Fitness</i>	Waktu (mili detik)
10	2 3 1	3201	70
20	2 3 1	3201	110
30	2 3 1	3201	116
40	2 3 1	3201	118

50	2	3	1	3201	120
60	2	3	1	3201	121
70	2	3	1	3201	125
80	2	3	1	3201	128
90	2	3	1	3201	130
100	2	3	1	3201	131

Dari tabel 4. dapat disimpulkan bahwa perubahan jumlah maksimum generasi akan mempengaruhi waktu proses algoritma genetika yang dilakukan. Pada kolom waktu (detik) terlihat bahwa semakin besar jumlah maksimum generasi, maka waktu yang dibutuhkan untuk melakukan proses algoritma genetika juga semakin besar.

Pengujian III

Pada pengujian III akan diamati pengaruh parameter probabilitas crossover dengan waktu proses. Pada pengujian III ini nilai parameter ukuran populasi, jumlah maksimum generasi, probabilitas mutasi dan probabilitas elitism akan diset tetap, sedangkan nilai parameter probabilitas crossover akan diset bervariasi mulai dari 0 sampai 1. Berikut ini disajikan tabel nilai parameter yang diset tetap yang digunakan pada pengujian III.

Tabel 4.5. Nilai Parameter Tetap pada Pengujian III

Parameter	Nilai Parameter
Jumlah Kandidat	3
Ukuran Populasi	10
Maksimum Generasi	70
Probabilitas Mutasi	0,2
Probabilitas Elistism	0,1

Tabel 6. Hasil Pengujian III

Probabilitas Crossover	Kromosom Terbaik	<i>Fitness</i>	Waktu (mili detik)
0,1	2 3 1	3201	90
0,2	2 3 1	3201	110
0,3	2 3 1	3201	111
0,4	2 3 1	3201	118
0,5	2 3 1	3201	122
0,6	2 3 1	3201	203
0,7	2 3 1	3201	205
0,8	2 3 1	3201	208
0,9	2 3 1	3201	210

Dari tabel 6. dapat disimpulkan bahwa perubahan probabilitas crossover mempengaruhi waktu proses algoritma genetika yang dilakukan. Pada kolom waktu (detik) terlihat bahwa

semakin besar probabilitas crossover, maka waktu yang dibutuhkan untuk melakukan proses algoritma genetika juga semakin meningkat.

Pengujian IV

Pada pengujian IV akan diamati pengaruh parameter probabilitas mutasi dengan waktu proses. Pada pengujian IV ini nilai parameter ukuran populasi, jumlah maksimum generasi, probabilitas crossover dan probabilitas elitism akan diset tetap, sedangkan nilai parameter probabilitas mutasi akan diset bervariasi mulai dari 0 sampai 1. Tabel 7 disajikan nilai parameter yang diset tetap yang digunakan pada pengujian IV.

Tabel 7. Nilai Parameter Tetap pada Pengujian IV

Parameter	Nilai Parameter
Jumlah Kandidat	3
Ukuran Populasi	10
Maksimum Generasi	70
Probabilitas Crossover	0,8
Probabilitas Elistism	0,1

Tabel 8. Hasil Pengujian IV

Probabilitas Mutasi	Kromosom Terbaik	<i>Fitness</i>	Waktu (mili detik)
0,1	2 3 1	3201	66
0,2	2 3 1	3201	110
0,3	2 3 1	3201	113
0,4	2 3 1	3201	118
0,5	2 3 1	3201	122
0,6	2 3 1	3201	200
0,7	2 3 1	3201	201
0,8	2 3 1	3201	206
0,9	2 3 1	3201	221

Dari tabel 8. dapat disimpulkan bahwa perubahan probabilitas mutasi mempengaruhi waktu proses algoritma genetika yang dilakukan sama seperti pada perubahan probabilitas crossover. Pada kolom waktu (detik) terlihat bahwa semakin besar probabilitas mutasi, maka waktu yang dibutuhkan untuk melakukan proses algoritma genetika juga semakin meningkat.

Pengujian V

Pada pengujian V akan diamati pengaruh parameter probabilitas elitism dengan waktu proses. Pada pengujian V ini nilai parameter ukuran populasi, jumlah maksimum generasi, probabilitas crossover dan probabilitas mutasi akan diset tetap, sedangkan nilai parameter probabilitas elitism akan diset bervariasi mulai dari 0 sampai 1. Berikut ini disajikan tabel nilai parameter yang diset tetap yang digunakan pada pengujian V.

Tabel 9. Nilai Parameter Tetap pada Pengujian V

Parameter	Nilai Parameter
Jumlah Kandidat	3
Ukuran Populasi	10
Maksimum Generasi	70
Probabilitas Crossover	0,8
Probabilitas Mutasi	0,2

Tabel 10. Hasil Pengujian V

Probabilitas Elistism	Kromosom Terbaik	<i>Fitness</i>	Waktu (mili detik)
0,1	2 3 1	3201	90
0,2	2 3 1	3201	110
0,3	2 3 1	3201	111
0,4	2 3 1	3201	118
0,5	2 3 1	3201	122
0,6	2 3 1	3201	203
0,7	2 3 1	3201	205
0,8	2 3 1	3201	208
0,9	2 3 1	3201	220

Dari tabel 10. dapat disimpulkan bahwa perubahan probabilitas elitism mempengaruhi waktu proses algoritma genetika yang dilakukan, sama seperti pada perubahan probabilitas crossover dan pada perubahan probabilitas mutasi. Pada kolom waktu (detik) terlihat bahwa semakin besar probabilitas elitism, maka waktu yang dibutuhkan untuk melakukan proses algoritma genetika juga semakin meningkat.

Pengujian VI

Setelah dilakukan pengujian I, II, III, IV dan V untuk mengetahui hubungan antara masing-masing parameter algoritma genetika terhadap waktu proses, selanjutnya akan dilakukan pengujian VI untuk mengetahui *fitness* optimal dengan nilai parameter yang tetap. Akan dilakukan beberapa proses untuk mendapatkan nilai *fitness* optimal.

Tabel 11. Parameter pada Pengujian VI

Parameter	Nilai Parameter
Jumlah Kandidat	3
Ukuran Populasi	20
Maksimum Generasi	70
Probabilitas Crossover	0,8
Probabilitas Mutasi	0,2

Probabilitas Elistism	0,1
-----------------------	-----

Hasil dari pengujian VI disajikan dalam tabel 12. sebagai berikut:

Tabel 12. Hasil Pengujian VI

Proses ke	Kromosom Terbaik	<i>Fitness</i>	Waktu (mili detik)
1	2 1 3	3199	64
2	2 3 1	3201	70
3	2 1 3	3199	77
4	2 3 1	3201	90
5	2 3 1	3201	100
6	2 3 1	3201	101
7	2 3 1	3201	104
8	2 3 1	3201	107
9	2 3 1	3201	110
10	2 3 1	3201	124

Dari tabel 12. didapat *fitness* terbesar adalah 3201 yang dimiliki 2 kromosom yaitu kromosom 2 3 1. Frekuensi kemunculan *fitness* pada pengujian VI terlihat seperti pada tabel 13.

Tabel 13. Frekuensi Kemunculan *Fitness* pada Pengujian VI

<i>Fitness</i>	Kromosom	Frekuensi Kemunculan <i>Fitness</i>
3201	2 3 1	8
3199	2 1 3	2

Sehingga dapat disimpulkan bahwa *fitness* optimal yang dicapai oleh algoritma genetika adalah 231 dengan urutan pemain seperti tampak pada tabel 14 sama dengan hasil pengujian

pada sistem menggunakan metode algoritma genetika pada gambar 8 yang sebelumnya telah dijelaskan pada gambar 6.

Tabel 14. Urutan Pemain Sesuai dengan *Fitness* Optimal

Urutan	Pemain
1	Anta
2	Ian
3	Vino



Gambar 8 Perhitungan Hasil Pengurutan Pemain

5. Kesimpulan

Setelah melakukan analisis, perancangan, implementasi beserta pengujian, maka dapat diperoleh kesimpulan sebagai berikut:

1. Sistem yang dibangun mampu membantu seorang pelatih merubah cara penilaiannya dalam menyeleksi pemain, agar dapat menilai dari yang bersifat subjektif menjadi lebih objektif lagi.
2. Sistem yang dibangun dapat menentukan posisi yang ideal untuk seorang pemain, dapat menentukan posisi terbaik dari masing-masing posisi dari sejumlah pemain.
3. Hasil dari Algoritma Genetika pada sistem yang dibangun sesuai dengan hasil perhitungan manual.

Selain kesimpulan, penelitian ini juga memiliki saran yang dapat digunakan untuk pengembangan dan penelitian selanjutnya yang terkait dengan penelitian ini. Saran tersebut antara lain:

1. Diharapkan keberadaanya bisa lebih *user friendly* dan lebih bermanfaat bagi kemajuan sepak bola.
2. Hasil yang diperoleh tidak bisa dihitung dalam matematika pada pengujian pertandingan.

6. Referensi

- [1] Daihani, D.U., Komputerisasi Pengambilan Keputusan, Jakarta, PT Gramedia, 2001.
- [2] Davis, L., Handbook of Genetic Algorithms, New York, Van Nostrand Reinhold, 1991.
- [3] Gen, M., Cheng, R., Genetic Algorithms and Engineering optimization, Japan, Ashikaga Institute of Technology, 1999.
- [4] Jogiyanto, Analisis Dan Desain, Yogyakarta, Andi, 2005.
- [5] Kusumadewi S., Purnomo H., Penyelesaian Masalah Optimasi dengan Teknik-teknik Heuristik, Yogyakarta, Graha Ilmu, 2005.
- [6] (26 April 2012) Prayogo, M, Sistem Pendukung Keputusan Untuk Menentukan Penerima Beasiswa Menggunakan Algoritma Genetika Dengan Metode Fuzzy Logic Sebagai Inisialisasi Awal (Studi Kasus: Mahasiswa Fakultas Pendidikan Matematika dan Ilmu Pengetahuan Alam Universitas Pendidikan Indonesia),[Online],
<http://journal.upi.ac.id/index.php/Snati/article/viewFile/1073/998/>
- [7] Suryadi K., Ramdhani M.A., Sistem Pendukung Keputusan, Bandung, PT Remaja Rosdakarya, 2002.
- [8] Suyanto, Algoritma Genetika dalam Matlab, Yogyakarta, Andi, 2010.
- [9] (3 Juni 2012) Syamsudien, Aries, Pengenalan Algoritma Genetik. [Online].Tersedia:
<http://ilmukomputer.org/2006/08/25/pengenalan-algoritma-genetik/>
- [10] Turban, *Decision Support System and Intelligent System* (Sistem Pendukung Keputusan dan Sistem Cerdas), Yogyakarta: Andi, , 2005